

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (currently amended): A method of operating a computer, the method comprising:  
for data structures in a set of data structures, as unloaded data structures are needed  
during runtime,  
receiving a data structure from a first memory, the data structure including one or  
more sets of instructions and one or more constants;  
storing instructions from the data structure in a first portion of a second memory,  
the second memory comprising RAM;  
storing constants from the data structure in a second portion of the second  
memory if and only if the respective constant has not been stored in the second portion of  
the second memory,  
modifying indexes in instructions that reference the constants to correspond to the  
respective locations of the constants in the second portion of the second memory, and  
reading and executing at least some instructions from the data structure from the  
RAM.
2. (original): The method of claim 1, wherein the data structures comprise classes.
3. (currently amended): The method of claim 1, wherein the data structures  
comprise Java classes of an object-oriented computer language that uses bytecode.

4. (original): The method of claim 1, wherein the sets of instructions comprise methods.
5. (currently amended): The method of claim 1, wherein the sets of instructions comprise Java methods of an object-oriented computer language that uses bytecode.
6. (original): The method of claim 1, wherein the constants from the data structure comprise a constant pool.
7. (original): The method of claim 1, wherein receiving the data structure from a first memory comprises receiving the data structure from a server over the Internet.
8. (original): The method of claim 1, wherein modifying indexes in instructions includes replacing respective indexes with larger indexes and wherein the method further includes calculating addresses associated with branch instructions.
9. (currently amended): A method of operating a computer, the method comprising:  
for classes in a set of classes, as unloaded classes are needed during runtime,  
receiving a class from a class file, the class including one or more methods and one or more constants;  
storing instructions from the class in a first portion of a memory;  
storing constants from the class in a second portion of the memory if and only if the respective constant has not been stored in the second portion of the memory,  
modifying indexes within methods that reference the constants to correspond to the respective locations of the constants in the second portion of the memory, and

executing from the memory at least some instructions from the class before receiving another class from the class file.

10. (currently amended): The method of claim 9, wherein the classes comprise Java classes of an object-oriented computer language that uses bytecode.

11. (original): The method of claim 9, wherein the memory comprises RAM.

12. (original): The method of claim 9, wherein receiving the class from a class file comprises receiving the class from a server over the Internet.

13. (original): The method of claim 9, wherein modifying indexes within methods includes replacing respective indexes with larger indexes and wherein the method further includes calculating addresses associated with branch instructions.

14. (currently amended): The method of ~~claim 9~~ claim 13, wherein the respective indexes each comprise 8 bits and the larger indexes each comprise 16 bits.

15. (original): The method of claim 9, wherein the constants comprise strings.

16. (currently amended): A computer system comprising:

a memory;

a virtual machine;

first logic that, after the virtual machine has been started, for classes in a set of classes,

receives a class from a class file, the class including one or more methods and one

or more constants;

stores instructions from the class in a first portion of the memory;

stores constants from the class in a second portion of the memory if and only if the respective constant has not been stored in the second portion of the memory, and modifies indexes within methods that reference the constants to correspond to the respective locations of the constants in the second portion of the memory; and second logic that executes methods stored in the memory; wherein the memory, the first logic, and the second logic are coupled locally.

17. (currently amended): The computer system of claim 16, wherein the classes comprise Java classes of an object-oriented computer language that uses bytecode.

18. (original): The computer system of claim 16, wherein the constants from the class comprise a constant pool of the data structure.

19. (original): The computer system of claim 16, wherein the memory comprises RAM.

20. (original): The computer system of claim 16, wherein receiving the class from a class file comprises receiving the class from a server over the Internet.

21. (original): The computer system of claim 16, wherein modifying indexes within methods includes replacing respective indexes with larger indexes and wherein the method further includes calculating addresses associated with branch instructions.

22. (currently amended): The computer system of ~~claim 16~~ claim 21, wherein the respective indexes each comprise 8 bits and the larger indexes each comprise 16 bits.

23. (currently amended): The computer system of claim 16, wherein the first and second logic comprises computer readable code means loaded into a RAM.